Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 94 (2016) 453 – 458

Procedia Computer Science

The 2nd International Workshop on Future Information Security, Privacy and Forensics for Complex systems (FISP-2016)

Enhancing Relational Database Security by Metadata Segregation

Devanshu Trivedi, Pavol Zavarsky, Sergey Butakov

Department of Information Systems Security and Assurance Management
Concordia University of Edmonton
Edmonton T5B 4E4, Alberta, Canada
dtrivedi@student.concordia.ab.ca, {pavol.zavarsky, sergey.butakov}@concordia.ab.ca

Abstract

Although many prominent Relational Database Management Systems provides inbuilt security controls and mechanisms, the information resided in the data-store are at great risk. This research aims to reduce the risk of unauthorized data access by providing an extra layer of security. This research proposes a novel method for incorporating information security while designing the relational database by segregating the information on the basis of its sensitivity level and creating referential integrity constraints dynamically at run time. Different techniques to identify and quantify sensitive attributes and restructuring database architecture have been discussed for the proposed approach. The primary keys of the restructured tables and most critical information attributes were secured using Transparent Data Encryption utility provided by Oracle 11g to prohibit illegitimate use of information. The performance of the proposed architecture was evaluated with 1,000,000 records which shows that by increasing the number of records, the response time of Select statement increased dramatically whereas it increased gradually for Insert, Update and Delete operations.

Peer-review under responsibility of the Conference Program Chairs

Keywords: constraint; integrity; Oracle 11g; primary key; procedure; synonym; view; foreign key

1. Introduction

There are mainly three phases where information must be secured. Firstly, in any application where data is being processed. Furthermore, while information travels on network channels. Finally, in data stores where the information reside for future usage. Various database management systems provide different controls for securing the data, once it is stored in database.

* Corresponding author. Tel.: +1-780-604-8258; fax: +1-780-378-8460.
E-mail address: dtrivedi@student.concordia.ab.ca

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.08.070

Oracle 11g provides user access management controls, database redaction policies, data encryption and integrity, wallet manager, auditing [9,10].Whereas Microsoft SQL server relational database is limited in security features with authentication, roles and access management, ownership and user schema management, authorization and permission on objects, encryption[13]. There are many manuals and white papers available from the vendors and third parties on how to secure the data by implementing these controls. But not so many papers provide information on how to integrate security features into relational database at the time of database schema design.

Many ground level concepts of Relational Databases provide low level security to the schema design such as views, synonyms, managing user profiles and access levels. One approach to secure information from unauthorized access is to create views on tables, where user interact with views and if the operation is found legitimate, the changes can be saved in original tables. These views can be created by selecting certain columns and/or rows from a table including permissions which creates restricted access to the information. Also, the Database Administrator can hide schema name or fully qualified object name from database consumers using synonyms.

Another approach is to segregate database columns with respect to their information sensitivity level and creating referential integrity constraints at run time which can make these columns isolated from each other. This mechanism would have one or more attributes/columns in the same table. The isolated tables can be created on different tablespace and placed on different geographical networks/servers. If an attacker gets an access to one table, only partial information will be available which is not sufficient for a successful attack. Such mechanism can decrease the potential risk of unauthorized data access and data-theft.

The information in database can be protected by securing its respective metadata. In a relational database, metadata can be schema name (table, procedure, function, trigger), column name or constraint name, type and definition. The logical alignment of these schemas can also be a metadata. If information from one table is compromised, the other tables which are logically related to that table are more susceptible as an attacker can jump from one table to another using referential integrity constraints. Other tables can be saved from an attacker by protecting such logical relations.

To simulate the above approach, healthcare is an appropriate industry where personal information of people must be protected. The research work was carried by implementation of a system for healthcare in Oracle 11g, where patient's personal data like social insurance number and health insurance number were secured. The system was developed by following object oriented analysis and design principles.

2. Related Works

Although very few papers recommend how to align information security at the time of database schema design, numerous research projects have been done on segregation of datasets. The separation of the data can be achieved in number of ways. One approach is to divide tables into several number of rows or columns. The second approach is to divide data sets based on their usability[4]. Another approach for segregation of data can be based on data ownership[12].

In[1], the concept of dividing the data on the basis of its sensitivity level and creation of logical relationship at run time to secure sensitive information was discussed which has some issues like concurrency, availability, audit and log management. Researchers in[2] have developed a framework to trade and exchange sensitive information on a shared network by calculating risk value for considered information. Risk based calculation for sensitive information was developed to exchange critical information between allies in a war scenario. Authors in[3], developed a method for data leakage prevention for cloud databases where they try to segregate the data over the cloud on the basis of the relationship between attributes. They developed a code scheme algorithm to store each sensitive data into scattered tables with respective code and shared the same code on client site.

In[5], author introduced a method for data recovery, at any point of time for comprehensive versioning systems using indexing called as Hierarchical Spatial-Temporal Indexing Method implemented by dividing the time domain in different partitions and in respect to the frequency of update operation on disk IOs. Whereas, in[8], author introduced a concept called point in time architecture which can be used to recover the data from the database at any given time. Usually, the previous state of data cannot be retrieved, once the operation is performed and new state of the data is committed. According to the concept, one can retrieve the state of the data at any point of time, before or after any operation. This database also can be used for auditing as data is saved at every time stamp.

3. Proposed Database Architecture

From section 2, with the profound concepts of segregation of database tables on information sensitivity levels and creation of logical relationship at run time, the proposed system workflow is described below. Once the tables are divided by the Database Administrator, the procedures to create and drop foreign key constraints can be developed. The execution of any DML statement is described below.

3.1 System Workflow

- DB engine receives a query request from a user with certain parameters.
- Predefined PL/SQL procedures resided in the DB engine creates logical relations between the tables at run time based on table names and column names passed as parameters.
- An insert entry is made into Point in Time table for creation of foreign key constraint(s).
- DB engine generates result after performing computation on the basis of the information available from different tables.
- The result will be saved (Commit) into respective tables and displayed to the user.
- The foreign key constraint(s) are deleted (Dropped), hence logical relationships between tables will no more exist.
- An insert entry is made into Point in Time table for deletion (dropping) of foreign key constraint(s).

By considering the following scenario, a patient taking medicines from a pharmacy, the two major entities involved here are PATIENT and PHARMACY. Such entities can be mapped to their corresponding tables with respective attributes as shown in Figure 1.

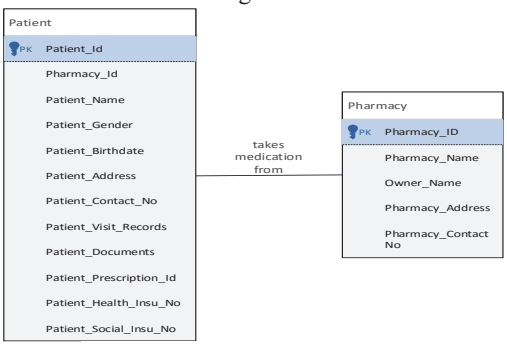

Figure 1. Entity Relationship diagram

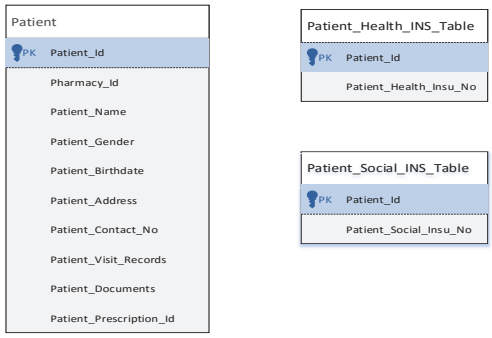

Figure 2. Segregated tables for Patient entity

Patient table has the personal information of any individual. Each attribute/column describes its own purpose with some sensitivity level. To implement the secure database approach, PATIENT table can be divided into smaller tables on attribute's sensitivity level as shown in Figure 2. The two columns "Patient_Health_Insu_No" and "Patient_Social_Insu_No" have the highest sensitivity level, so they have been separated into new tables namely "PATIENT_HEALTH_INS_TABLE" and "PATIENT_SOCIAL_INS_TABLE" respectively from "PATIENT_GENERIC_DETAILS_TABLE".

3.2 Reducing risk of unauthorized data access

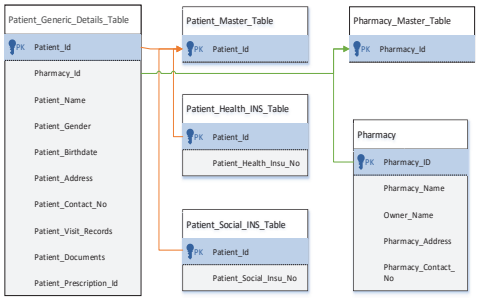

Figure 3. Referential Integrity to be created/deleted dynamically at the time of any DML operation

If an attacker gains unauthorized access of "PATIENT_SOCIAL_INS_TABLE" table, he only has the partial information of the patient, which is "Patient_Social_Insu_No" and its corresponding unique identifier "Patient_Id" that are encrypted with Transparent Data Encryption utility by Oracle 11g.

Logical relationships must be created between tables to maintain data integrity and consistency. But in this approach, the logical constraints will be created at run time, while any DML operation will be initiated by user. The referential integrity constraints always points child-parent relationships between two or more tables. If such referential integrity constraints are created at table schema design (in create table statements) and the attacker is able to track these foreign key constraints, then reverse engineering can be used to gain access of other tables.

These operations can be achieved by creating PL/SQL procedures dynamically, where parameters can be passed in terms of child_table_name, column_name and parent_table_name. Many transactions will occur as logical relations between the tables are created at run time such as foreign key creation/deletion. So to keep a track all activities made during each DML transaction, a Point in Time table was implemented for audit and log purpose. The Point in Time table contains fields like username – who has initiated the DML operation, time stamp – when the operation is performed, Creation/Deletion of Foreign key Constraint/s, DML operation type – Insert/ Update/Delete /Select, Child table name, Foreign key constraint column name and a Constraint name – unique identifier with format FK_ DD MM YY HH MIN SEC MILISEC.

4. Techniques and Discussion

In comparison to simple relational database design, this architecture gives more overhead to database engine for creation and deletion of foreign key constraints at run time.

4.1 Different techniques to identify and quantify sensitive attribute and restructuring database architecture

- Ranking:
  Give highest ranking to the most sensitive attribute and lowest to the least sensitive attribute.
  Example: Top Secret – 1, Secret – 2, Confidential – 3, Protected – 4
- Decomposition of attributes:
  The attribute can be divided into smaller sub-attributes. These sub-attributes can act as an individual attribute in segregated tables.
- Logical grouping:
  More attributes can be grouped into same tables, if they serve the same purpose and are less valuable.
  Example: Patient_Generic_Details_Table can be more fragmented in Patient_Biometric_Details_Table and Patient_Contact_Details_Table.

For the developed simulation, Ranking and Logical grouping techniques were used to identify and quantify sensitive attribute for segregation of tables. First, ranking technique was applied to patient and pharmacy tables' attributes, where Patient_Health_Insu_No and Patient_Social_Insu_No were ranked highest priority attributes amongst all other attributes. Two separate tables were created respectively as "PATIENT_HEALTH_INS_TABLE" and "PATIENT_SOCIAL_INS_TABLE" to place these crucial information. Secondly, the logical grouping of attributes were created for Patient and Pharmacy entities where less or no sensitive information was placed together in same table. The Patient_Generic_Details_Table contain the attributes which holds general information of any individual.

4.2 Transparent Data Encryption - TDE

The TDE is an inbuilt utility provided by Oracle 11g to encrypt and decrypt sensitive data. The TDE was applied on all the primary keys (Patient_Id, Pharmacy_Id) of different tables. The TDE was also used to secure the most critical information identified for any individual such as Patient_Health_Insu_No and Patient_Social_Insu_No.

5. Performance Measurement

The performance of the proposed relational database architecture was expected to be lowered while considering certain factors like size of the database – total records, degree of normalization used, volume – number of attributes, number of tables partitioned, integrity constraints created, validation constraints, total number of encrypted attributes

and index organization. The performance shown in Table II was measured with 1,000,000 patient and 200 pharmacy records. To provide more security to the system, the INSERT, UPDATE and DELETE statements are restricted to affect only one record at a time. The performance of SELECT statement is measured for all available patient records.

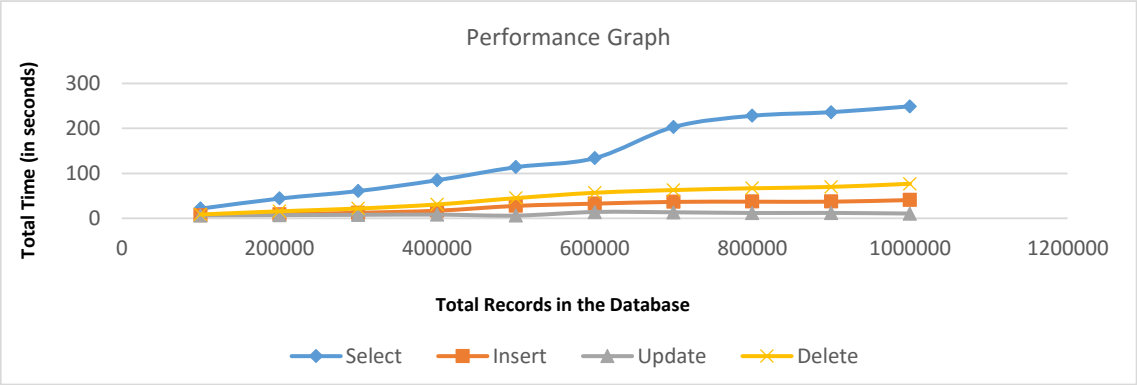

Figure 4. Performance Graph

Table 1. Performance measurement

Total no. of Records	Response Time (in seconds) for DML Operations			
	Select	Insert	Update	Delete
100000	21.95	7.1	6.4	8.74
200000	44.3	8.9	7.01	15.87
300000	61	12.38	7.67	22
400000	85	17	8.2	31
500000	114	27.9	6	45.1
600000	134	33	14	57
700000	203	37	13.31	63
800000	228	37.44	11.75	67
900000	236	37.42	11.91	70
1000000	249	41.15	10.5	77

Figure 4 is a scatter graph, indicating that the performance of DML operations is degrading with increasing number of records in the database. The graph also indicates that response time for INSERT, UPDATE and DELETE DML statements are less than 31 seconds for 400,000 records in the database whereas SELECT statement takes 85 seconds to display the same number of records. Overall, with the increase in number of records, the response time of Select statement increased dramatically while for INSERT, UPDATE and DELETE, it was gradual.

Different experiment strategies were followed to improve the performance by changing the code of SQL procedures. The Order By clause on Patient_Id was removed from SELECT statement, which increased the performance up to 22% which was before 321 seconds and decreased to 249 seconds to display 1,000,000 records.

6. Conclusion and Future Work

Additional efforts can be added to the design process to add security related features during schema development for an extra layer of security. With the segregation of metadata, referential integrity constraints can be created and

dropped at runtime which secure the data or devaluate the data if it is compromised. Though attacker gets an unauthorized access to the database system, scattered information between several tables will be useless and makes it extremely difficult to link the database objects to each other.

This approach can be implemented on different servers located on various geographical locations which requires extra effort to add more security. Though Oracle 11g creates a default Index on any primary key column, an additional Index can be created for faster retrieval of data. The Point in Time table can be customized in a more useful manner to record the data status before any operation which can be helpful for audit trails. The Point in Time table will act as a repository which contains all the records of the previously performed DML operations. So the Point in Time table is also a critical object of the proposed architecture which must be secured by placing it on different server.

The proposed approach can be further enhanced by placing a Query Sanitizer in between the user application interface and the database to reduce the risk of SQL injection attacks. The Query Sanitizer can be a SQL procedure to verify the parameters passed along with the DML operation which will change the proposed system workflow at step 2. When the user initiate the DML operation, it will check the legitimacy of the parameters. If the parameters are acceptable, then it will redirect to step 3 in proposed system workflow, else it should terminate the transaction immediately.

References

1. Subashini S and Dr. Kavitha V, “A Metadata Based Storage Model for Securing Data in Cloud Environment”, Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011, Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6079468
2. Mudhakar Srivatsa, Pankaj Rohatgi, Shane Balfe and Steffen Reidt from IBM TJ. Watson Research Center, “Securing Information Flows: A Metadata Framework”, Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on Sept. 29 2008-Oct. 2 2008, Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4660114
3. Ben Omran, O.M. Panda B, “A New Technique to Partition and Manage Data Security in Cloud Databases”, Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference on 8-10 Dec. 2014, Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7038803
4. Rasmus Resen Amossen, IT University of Copenhagen, “Vertical partitioning of relational OLTP databases using integer programming”, Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on 2010, Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5452739
5. Yonghong Sheng, Dan Xu, Dongsheng Wang, “A High Effective Indexing and Retrieval Method Providing Block-Level Timely Recovery to Any Point-In-Time”, Networking, Architecture and Storage (NAS), 2010 IEEE Fifth International Conference, 2010, Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5575630
6. Dr. L. Arockiam S. Monikandan, “Efficient Cloud Storage Confidentiality to Ensure Data Security”, Computer Communication and Informatics (ICCCI), 2014, Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6921762
7. Jegadeeswari, S.; Dinadayalan, P.; Gnanambigai, N., “A Neural Data Security Model – Ensure High Confidentiality and Security in Cloud Data Storage Environment”, Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference, Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7275642
8. Arthur Fuller, “Database Design: A Point in Time Architecture”, Publish in: Online; Available at: https://www.simpletalk.com/sql/database-administration/database-design-a-point-in-time-architecture/
9. “Oracle Database Security Guide 11g Release 2 (11.2)”, Available at: https://docs.oracle.com-/cd/E11882_01/network.112/e36292.pdf
10. “Oracle Database Advanced Security Administrator's Guide 11g Release 2 (11.2)”; Available at: https://docs.oracle.com/cd/E11882_01/network.112/e40393.pdf
11. “SQL server best practices - implementation of Database object schemas” – Article by Microsoft Available at: https://technet.microsoft.com/enus/library/dd283095%28v=sql.100%29.aspx
12. “Data partitioning Guidance” – Article by Microsoft; Available at: https://msdn.microsoft.-com/en-us/library/-dn589795.aspx
13. “Overview of SQL server security” – Article by Microsoft; Available at: https://msdn.microsoft.com/enus/library/bb669078%28v=vs.110%29.aspx
14. “An Overview of Alberta’s Electronic Health Record Information System” Available at: http://www.albertanetcare.ca/documents/An_Overview_of_Albertas_ERHIS.pdf
15. “To generate large volumes of custom data in specified data format for testing activity“– Online Resource, www.generatedata.com